DIGITAL SIGNAL PROCESSING LABORATORY

EXPERIMENT-3 – DTMF ENCODER/DECODER

Group:- 32

Authors:-

Jaya Kishnani (20EC30020)

Gunjan Shekhar(20EC10032)

TA:- Abhishek Singh

Aim:

- Study and analysis of DTMF encoder / decoder.
- Implementation of Band Pass Filters.

Theory:

DTMF is a signalling system for identifying the keys pressed on a DTMF keypad. The frequencies corresponding to each key on the keypad can be represented in the form of a matrix.

Hz	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

(A telephone keypad and the DTMF frequencies for each column and row)

When the key is pressed the sinusoids of the corresponding frequencies are summed to generate an input signal.

<u>DTMF Decoder</u>-

This signal is then passed through a filter bank consisting of eight band pass filters. Each filter passes only one of the eight DTMF frequencies. The values of magnitude of two of the filters would be high compared to others. These two filters will signify the two frequencies present in the input signal. Using this and the DTMF frequency matrix, the pressed key can be decoded.

The Band Pass Filter is realised by defining the impulse response of L-point FIR filter-

$$h[n] = \beta\cos(\hat{w}c*n) \text{ where } 0 \leq n < L.$$

where L is the filter length, wc is the center frequency determining the location of the passband frequency and β is the term corresponding to the passband gain.

Observations and results:

Code-

```
f_mat = [697 770 852 941;
    1209 1336 1477 1633]; %Matrix containing the eight DTMF frequencies
L = 32;
fs = 4000; %sampling frequency
T = 50;
s = input("Enter keys:"); %input is taken as an array of pressed keys
x = zeros(length(s),fs*T+1);
```

```
%Generating the signals for the corresponding pressed keys
for n = 1:length(s)
    %input_sig() function is called and
    %the pressed key s(n) is passed as parameter
    x(n,:) = input_sig(s(n),fs,T,f_mat);
end
```

```
%The signals generated by pressing keys is passed
%to the function dtmf_decoder()
[decoded_output] = dtmf_decoder(x,f_mat);
decoded_output = char(decoded_output)
```

```
function [h] = bpf(a,b,f_mat,L)
%this function creates an FIR band pass filter corresponding
%to the frequency f_mat(a,b)
    h = zeros(1,L);
    fs = 4000;
    B = 0.5;
    wc = 2*pi*f_mat(a,b);
    w = wc/fs;
    for n = 1:L
        h(n) = B*cos(w*(n-1));
    end
end


function [output] = dtmf_decoder(x,f_mat)
%this function decodes the array of DTMF signals
% and gives an array of decoded keys as output
    L = 32;
    [no_of_inputs,~] = size(x);
    output = zeros(1,no_of_inputs);
    for i = 1:no_of_inputs
        y1 = filter(bpf(1,1,f_mat,L),1,x(i,:));
        Y1 = max(y1);
```

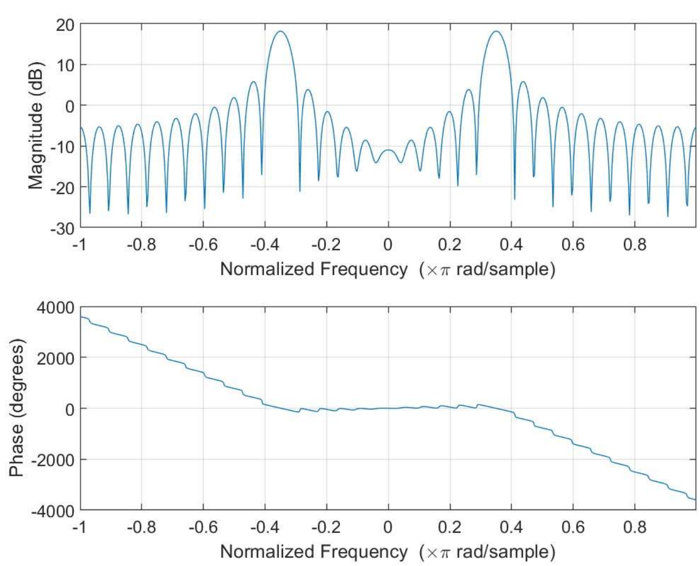
```
        y2 = filter(bpf(1,2,f_mat,L),1,x(i,:));
        Y2 = max(y2);
        y3 = filter(bpf(1,3,f_mat,L),1,x(i,:));
        Y3 = max(y3);
        y4 = filter(bpf(1,4,f_mat,L),1,x(i,:));
        Y4 = max(y4);
        y5 = filter(bpf(2,1,f_mat,L),1,x(i,:));
        Y5 = max(y5);
        y6 = filter(bpf(2,2,f_mat,L),1,x(i,:));
        Y6 = max(y6);
        y7 = filter(bpf(2,3,f_mat,L),1,x(i,:));
        Y7 = max(y7);
        y8 = filter(bpf(2,4,f_mat,L),1,x(i,:));
        Y8 = max(y8);
        fmax = [Y1 ,Y2 ,Y3, Y4];
        Fmax = [Y5, Y6, Y7, Y8];
        [~, index1] = max(fmax);
        [~, index2] = max(Fmax);
        detected_op = ['1' '2' '3' 'A';
            '4' '5' '6' 'B';
            '7' '8' '9' 'C';
            '*' '0' '#' 'D'];
        output(i) = detected_op(index1,index2);
    end
end


function x = input_sig(s,fs,T,f_mat)
%this function generates the DTMF signal corresponding to the pressed key
    t = 0:1/fs:T;
    if s=='1'
        x = 3*sin(2*pi*f_mat(1,1)*t)+4*sin(2*pi*f_mat(2,1)*t);
    end
    if s=='2'
        x = 3*sin(2*pi*f_mat(1,1)*t)+4*sin(2*pi*f_mat(2,2)*t);
    end
    if s=='3'
        x = 3*sin(2*pi*f_mat(1,1)*t)+4*sin(2*pi*f_mat(2,3)*t);
    end
    if s=='A'
        x = 3*sin(2*pi*f_mat(1,1)*t)+4*sin(2*pi*f_mat(2,4)*t);
    end
    if s=='4'
        x = 3*sin(2*pi*f_mat(1,2)*t)+4*sin(2*pi*f_mat(2,1)*t);
    end
    if s=='5'
        x = 3*sin(2*pi*f_mat(1,2)*t)+4*sin(2*pi*f_mat(2,2)*t);
    end
    if s=='6'
        x = 3*sin(2*pi*f_mat(1,2)*t)+4*sin(2*pi*f_mat(2,3)*t);
    end
```

```
    if s=='B'
        x = 3*sin(2*pi*f_mat(1,2)*t)+4*sin(2*pi*f_mat(2,4)*t);
    end
    if s=='7'
        x = 3*sin(2*pi*f_mat(1,3)*t)+4*sin(2*pi*f_mat(2,1)*t);
    end
    if s=='8'
        x = 3*sin(2*pi*f_mat(1,3)*t)+4*sin(2*pi*f_mat(2,2)*t);
    end
    if s=='9'
        x = 3*sin(2*pi*f_mat(1,3)*t)+4*sin(2*pi*f_mat(2,3)*t);
    end
    if s=='C'
        x = 3*sin(2*pi*f_mat(1,3)*t)+4*sin(2*pi*f_mat(2,4)*t);
    end
    if s=='*'
        x = 3*sin(2*pi*f_mat(1,4)*t)+4*sin(2*pi*f_mat(2,1)*t);
    end
    if s=='0'
        x = 3*sin(2*pi*f_mat(1,4)*t)+4*sin(2*pi*f_mat(2,2)*t);
    end
    if s=='#'
        x = 3*sin(2*pi*f_mat(1,4)*t)+4*sin(2*pi*f_mat(2,3)*t);
    end
    if s=='D'
        x = 3*sin(2*pi*f_mat(1,4)*t)+4*sin(2*pi*f_mat(2,4)*t);
    end
end
```

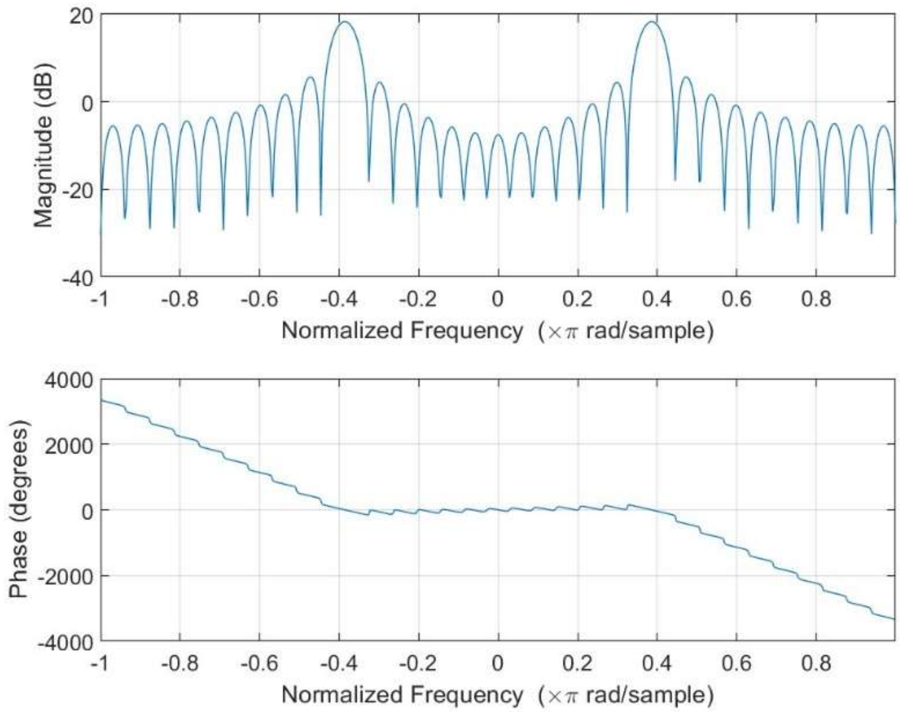
Plots-

L = 32

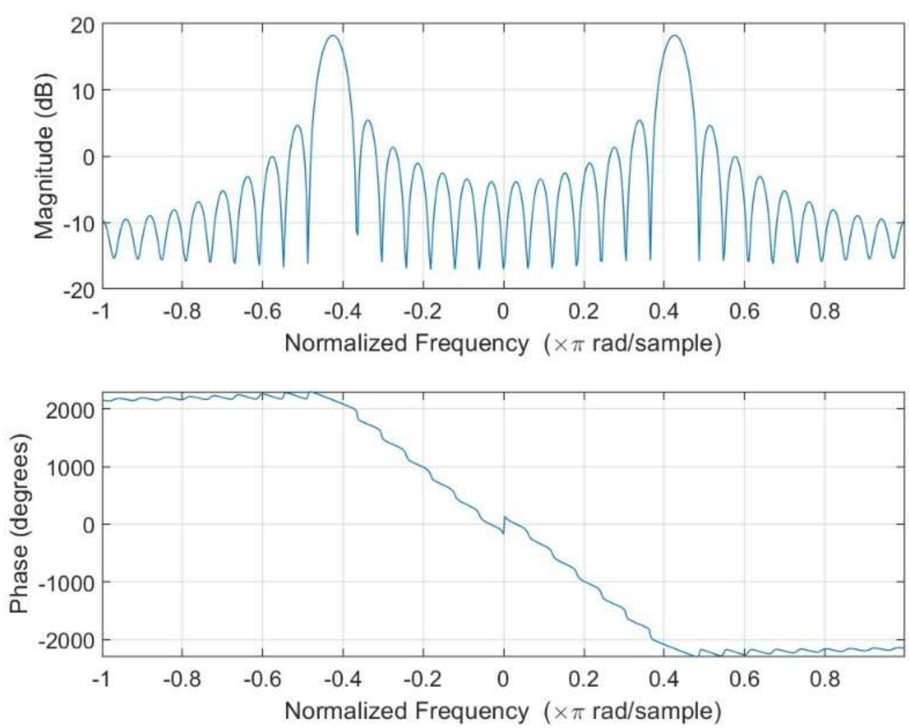
Frequency Response of BPF for fc = 697 Hz



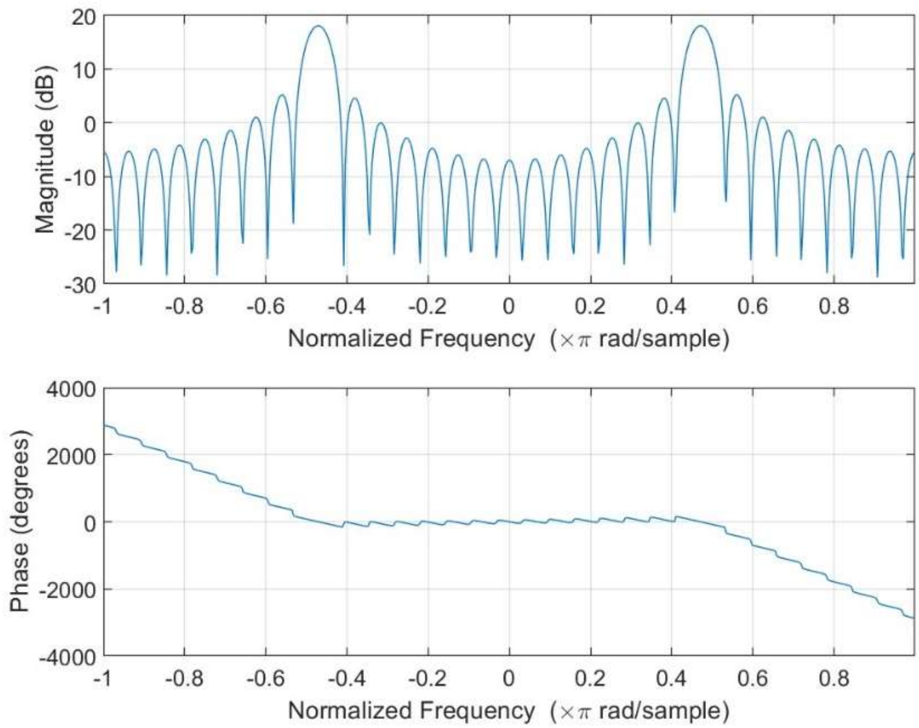
Frequency Response of BPF for fc = 770 Hz



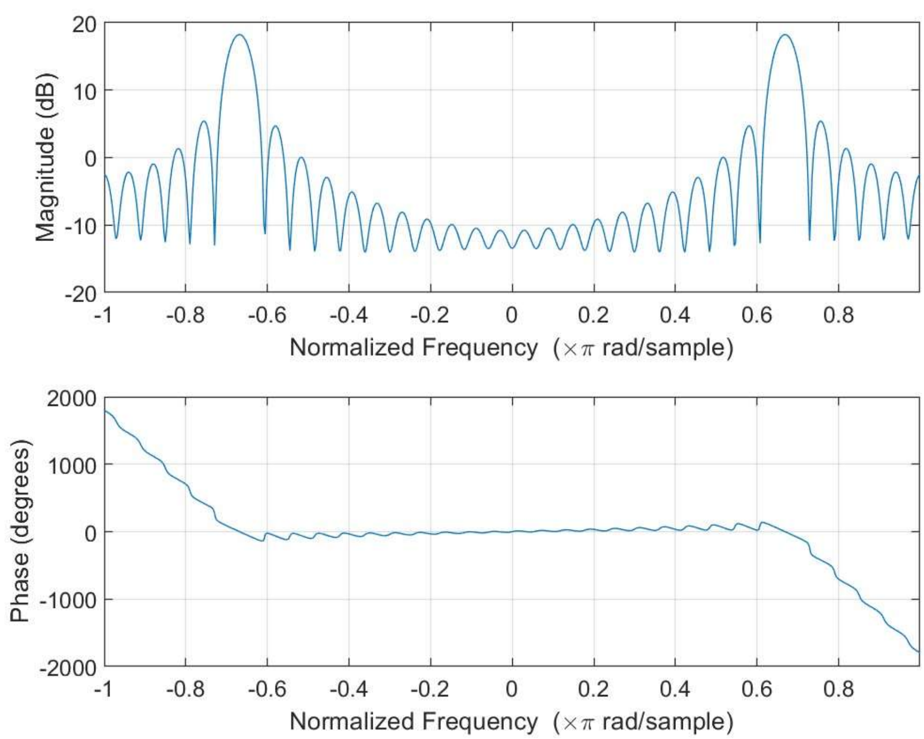
Frequency Response of BPF for fc = 852 Hz



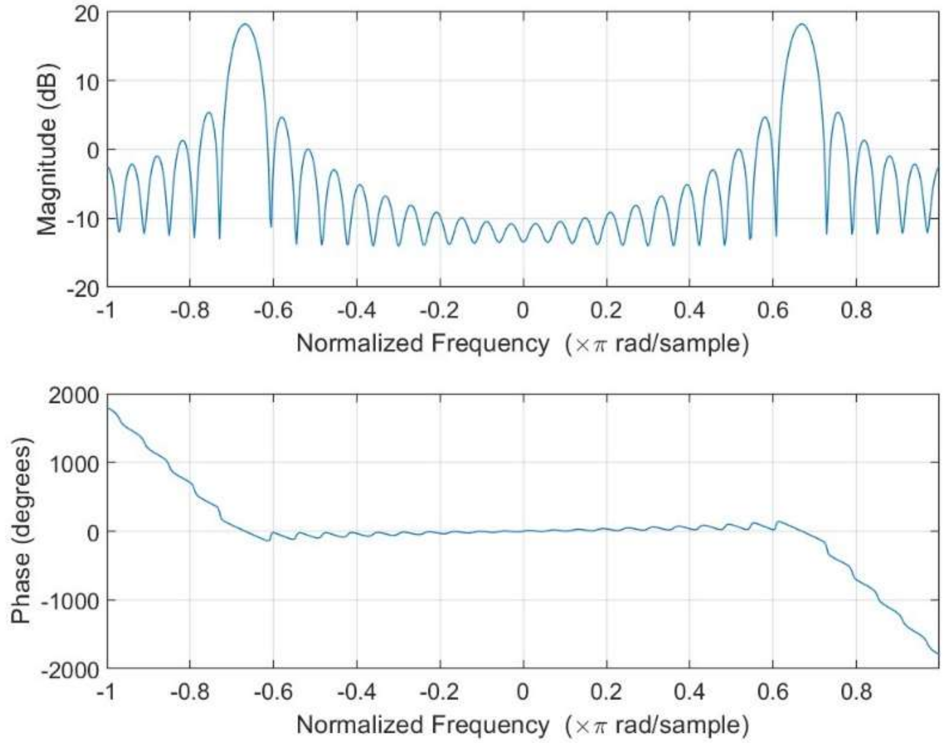
Frequency Response of BPF for fc = 941 Hz



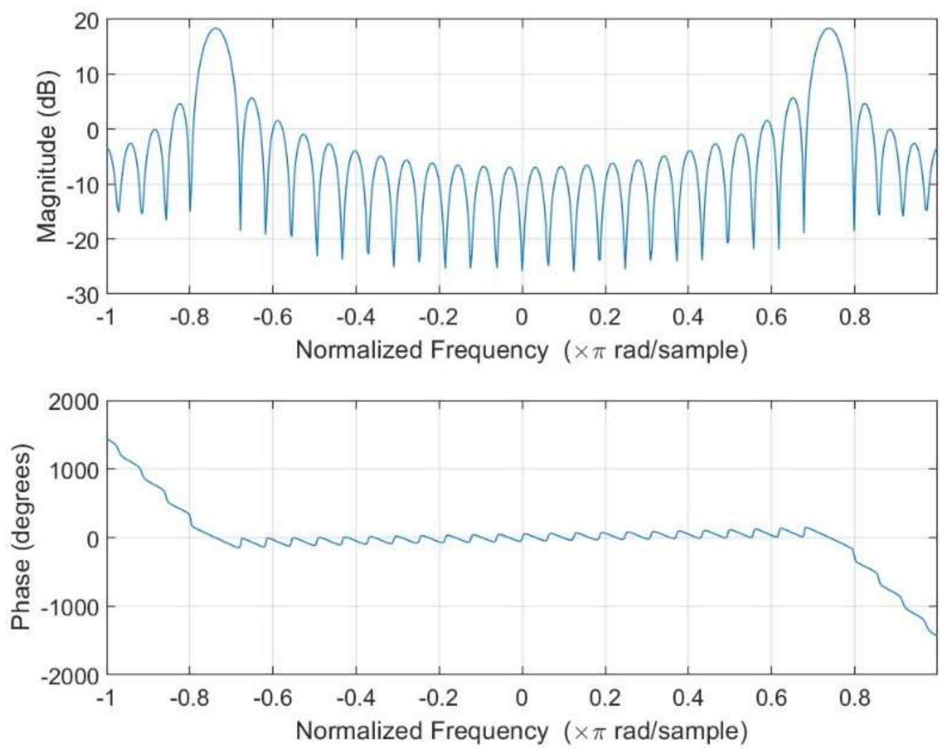
Frequency Response of BPF for fc = 1209 Hz



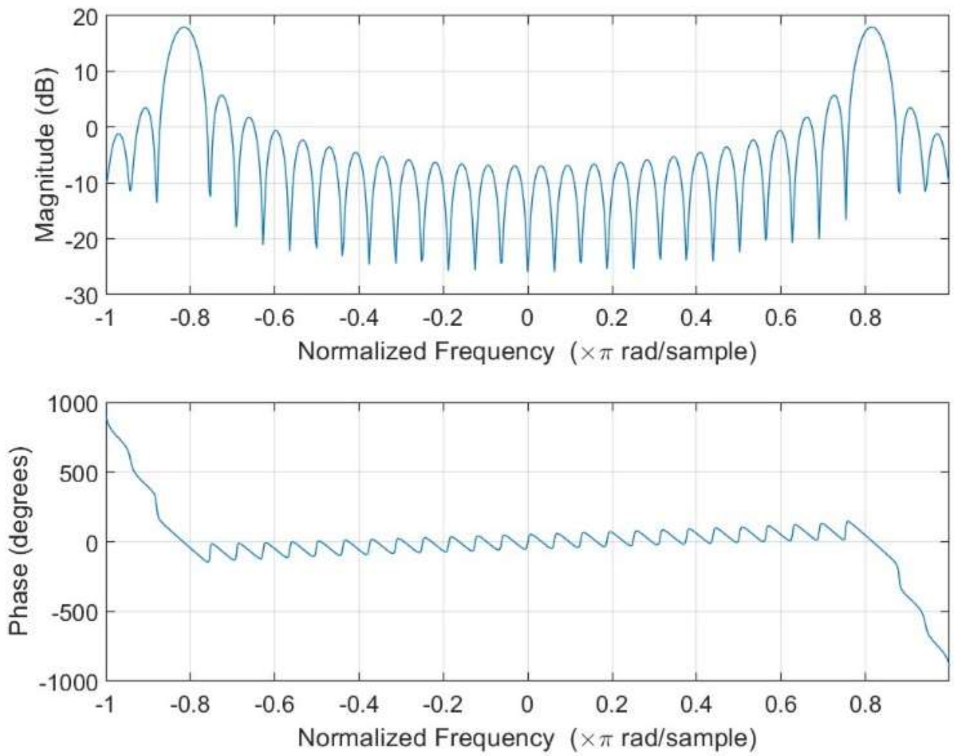
Frequency Response of BPF for fc = 1336 Hz



Frequency Response of BPF for fc = 1477 Hz



Frequency Response of BPF for fc = 1633 Hz



Example-

Entered input:

```
Command Window
Enter keys like ['' ''] :['2' '3' '5' '6' '1' '4' 'A' 'C' 'B' '8' 'D' '7' '*' '0' '9' '#']
fx >>
```

Received output:

```
decoded_output = '235614ACB8D7*09#'
```

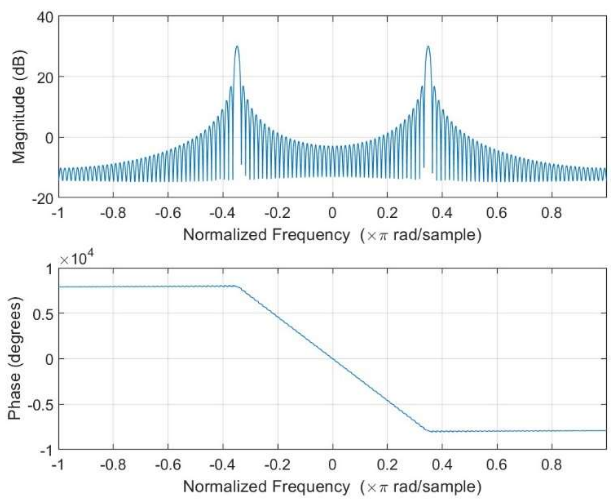
Discussion:

- The values of the pressed keys are taken as an input array and then given to the DTMF encoder which turns it into a signal containing the summation of two sine waves of corresponding frequencies. This is then passed to DTMF decoder which uses a filter bank to decode and give the pressed key as an output.

- When the signal is passed through the filter bank, the signal is passed through the eight FIR bandpass filters and the maximum value of the output of these filters is recorded. If the frequency corresponding to the filter is present in the signal, the maximum value recorded is comparatively higher compared to other output of other filters.

- The function h(n) = Bcos($\overline{wc}$*n) is used to realise the band pass signal. It can be seen from its frequency response that the passband of this signal is around the center frequency, wc. And so only the frequencies very close or equal to wc can pass through this filter and give a significant output.

- When the value of L is increased, the frequency response of the FIR filter becomes smoother and its bandwidth decreases. A sharper peak is obtained at the center frequency.



Frequency response of BPF for fc = 697 Hz at L = 128

- Here, the max() value of the output of each filter is used to detect the frequencies present in the input signal. This detection can also be done using the sumsqr() function which basically calculates the energy of the function. If the signal does not contain the center frequency, the filter will suppress the energy of the signal because it only transmits the energy corresponding to its center frequency.